```
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.utils import to_categorical

# Load the MNIST dataset
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Normalize the input images
x_train = x_train / 255.0
x_test = x_test / 255.0

# One-hot encode the labels
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)

# Build the neural network model
model = Sequential([
    Flatten(input_shape=(28, 28)),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
model.fit(x_train, y_train, epochs=5, batch_size=32, validation_split=0.1)

# Evaluate the model
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'\nTest accuracy: {test_acc:.4f}')

# Save the model
model.save('mnist_digit_recognizer.h5')
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-kera
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
/usr/local/lib/python3.11/dist-packages/keras/src/layers/reshaping/flat
  super().__init__(**kwargs)
Epoch 1/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 12s 6ms/step - accuracy: 0.8650 - loss: (
Epoch 2/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step - accuracy: 0.9654 - loss: 0
Epoch 3/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 8s 5ms/step - accuracy: 0.9768 - loss: 0
Epoch 4/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step - accuracy: 0.9836 - loss: 0
Epoch 5/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step - accuracy: 0.9862 - loss: 0
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.9739 - loss: 0.0
WARNING:absl:You are saving your model as an HDF5 file via `model.save(

Test accuracy: 0.9769
```

```
import numpy as np
import pandas as pd
from tensorflow.keras.datasets import mnist

# Load dataset
(x_train, y_train), _ = mnist.load_data()

# Flatten the 28x28 images to 784 features
```

Release notes Cell ×

```
import tensorflow as tf
from tensorflow.keras.datasets import mnis
from tensorflow.keras.models import Sequen
from tensorflow.keras.layers import Dense,
from tensorflow.keras.utils import to_cate

# Load the MNIST dataset
(x_train, y_train), (x_test, y_test) = mni

# Normalize the input images
x_train = x_train / 255.0
x_test = x_test / 255.0

# One-hot encode the labels
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)

# Build the neural network model
model = Sequential([
    Flatten(input_shape=(28, 28)),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentro
              metrics=['accuracy'])

# Train the model
model.fit(x_train, y_train, epochs=5, batc

# Evaluate the model
test_loss, test_acc = model.evaluate(x_tes
print(f'\nTest accuracy: {test_acc:.4f}')

# Save the model
model.save('mnist_digit_recognizer.h5')
```

```
Downloading data from https://storage.google
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0u
/usr/local/lib/python3.11/dist-packages/kera
  super().__init__(**kwargs)
Epoch 1/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 12s 6ms/step
Epoch 2/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step -
Epoch 3/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 8s 5ms/step -
Epoch 4/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step -
Epoch 5/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step -
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - a
WARNING:absl:You are saving your model as an

Test accuracy: 0.9769
```

```
# Flatten the 28x28 images to 784 features
x_train_flat = x_train.reshape((x_train.shape[0], -1))

# Combine images and labels into a single DataFrame
df = pd.DataFrame(x_train_flat)
df.insert(0, 'label', y_train)

# Save to CSV
df.to_csv('mnist_train_data.csv', index=False)
```